P.O. BOX 177, BOX HILL, 3128. $2.00

MCCC INC NEWS

THE MONTHLY NEWSLETTER FROM THE FAMILY COMPUTER CLUB

Melbourne Commodore Computer Club Inc.

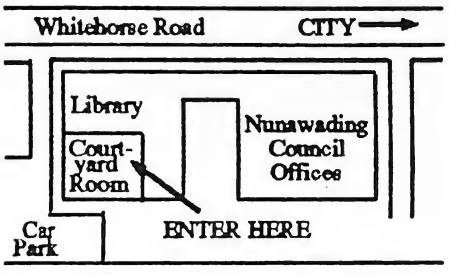
OCTOBER 1994

VIC 20 C16 PLUS 4 C64 C128 AMIGA

Melbourne Commodore Computer Club Inc.

Postal Address:
P.O. Box 177,
Box Hill,Vic. 3128.

Club meetings are held on the second Wednesday of each month at the Nunawading Civic Centre in the Courtyard Room.



Meetings begin at 7.30 p.m. Please make an effort to arrive on time so the meeting can begin with no delays.

All opinions expressed are those of the authors of individual articles and not necessarily those of the MCCC Inc.

Newsletter Editor:
Dorothy Millard
12 Venetian Court,
Croydon, Vic. 3136.
Telephone (03) 725-0682
Articles for the newsletter may be hardcopy (handwritten is equally acceptable), C64 or Amiga disk. All club members are invited to submit articles.

Secretary:
Robert Morrow
P.O. Box 651,
Templestowe, Vic. 3106.

DEADLINE FOR NEXT ISSUE
28th October, 1994.

In this Issue.....

MEMBERSHIP FEES - Visitors Welcome
$35 per year Family Membership
$3 VISITOR (family - includes FREE Newsletter)

BENEFITS OF MEMBERSHIP
* Monthly newsletter mailed to all members.
* Pedlar's trading table selling recycled computer wares and blank 3 1/2" and 5 1/4" disks at competitive prices.
* Magazine Library - Magazines are available for members to borrow.
* Access to the club's extensive C64 & Amiga PD libraries - club disks available at meetings for only $2 each.
* Help is available to assist in solving problems.
* Computers for members' use.
* Monthly demonstrations

OCTOBER DEMONSTRATIONS
Disks - How they work.
A look inside the Amiga
Programming for Beginners - How to get started.

COMMITTEE FOR 1994/95 CLUB YEAR

PRESIDENT	Bernie O'Shea
SECRETARY	Bob Morrow
SECRETARY'S ASSISTANT	Jim Davies
TREASURER	George Flanagan
EDITOR	Dorothy Millard
C64/128 LIBRARIAN	ShirleyYoung
AMIGA LIBRARIANS	Brett Eden & Darryl Hunter
PEDLAR/DISK SALES	
MAGAZINE LIBRARIAN	Grant Davies
PUBLICITY	Brett Eden
COMMITTEE MEMBER	Rob Jackson
COMMITTEE MEMBER	Leoni Parsons

The Editor's Bytes

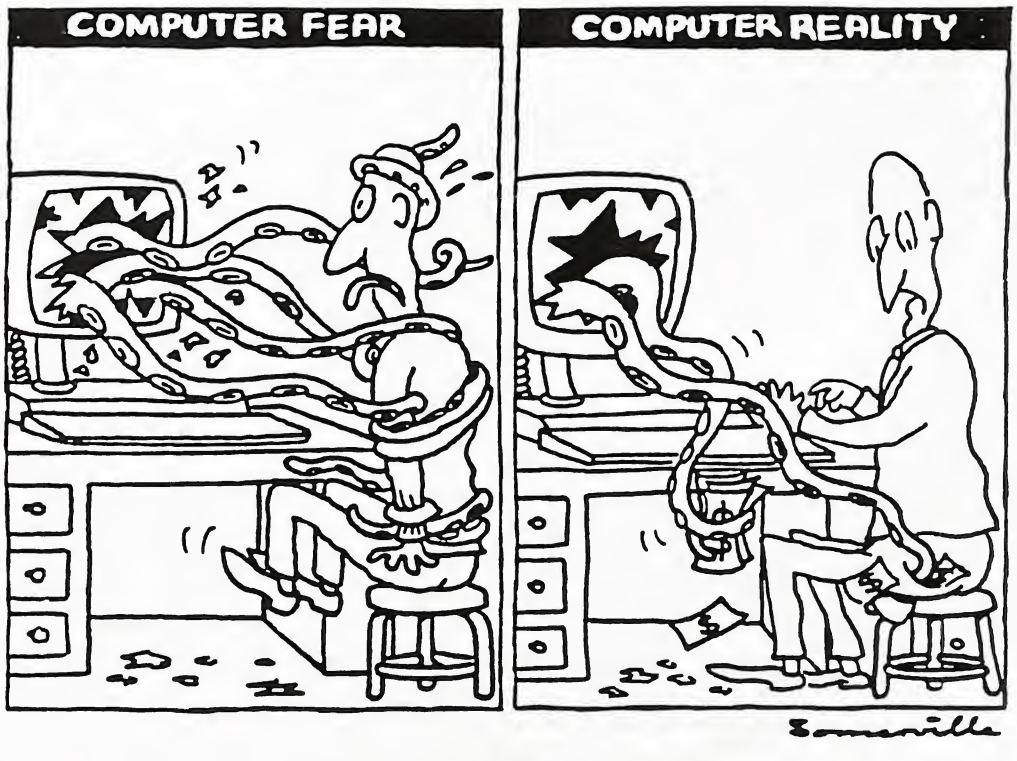
As the saying goes, it is all or nothing. Last month I had heaps of articles for inclusion in the newsletter but this month I have received nothing apart from Bernie's President's Report and the monthly disk. Even the Amiga article I was promised got lost along the way. Because of this you will find some reprints from other newsletters in this month's edition of MCCC News and I thank those clubs for allowing us to reprint their articles. Hopefully this situation will be remedied before the next edition goes to print and I will be overwhelmed by submissions - well I can hope!

I have been busy reviewing some games for an English adventure magazine using a Spectrum Emulator on my Amiga 500. It sure does look strange to see the Speccy screen come up. The emulator works reasonably well and does allow one to play a number of "classic" games and very good they are too. The games being reviewed by myself are text adventures so I don't know how well the emulator copes with graphics but I certainly had no problems. Even the MEMORY save worked well. The only problem I encountered using the emulator was because, being a fast typist, I found the program couldn't keep up with my typing and I had to consciously slow down.

If your Amiga suddenly stops working and the caps lock light flashes don't panic, it might be a simple loose connection. This has happened a couple of times to my 500 and the remedy was to unscrew the case and simply check the connections, in my case it was the plug to the keyboard which was loose.

Until next month.....

Dorothy



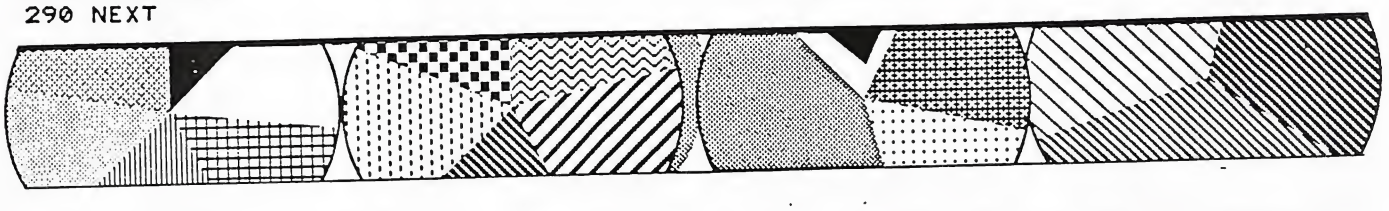
SCREEN COPY

by A. Packer

This is a nice simple routine that you can use within your own programs to provide a printed display of the screen. All the normal C64 chars. are catered for, including the reverse set.

To activate this program, just load and run.

```
100 REM *****SCREEN COPY*****
110 REM *BY A PACKER 14.2.88*
120 REM **********************
130 LET F=40
140 LET C$=""
150 FOR B=1024 TO 2023
160 LET A = PEEK(B)
170 POKE B,31
180 REM**POKE TO CHR$ CODE**
190 IF A=>0 AND A<=31 THEN LET C=A+64
200 IF A=32 THEN LET C=32
210 IF A=>33 AND A<=63 THEN LET C=A
220 IF A=>64 AND A<=95 THEN LET C=A+32
230 IF A=>97 AND A=<127 THEN LET C=A+64
240 IF A=>128 THEN GOTO 380
250 REM *** BUILD UP LINE ***
260 LET C$=C$+CHR$(C)
270 IF LEN(C$)=F THEN GOSUB 300
280 POKE B,A
290 NEXT
300 REM *****PRINT LINE******
310 OPEN 1,4
320 PRINT#1,C$
330 LET C$=""
340 LET F=40
350 CLOSE 1,4
360 IF B=2023 THEN END
370 RETURN
380 REM **INVERSE CHARACTERS**
390 LET C$=C$+CHR$(18):LET F=F+1
400 IF A=>128AND A<=159THEN LET C=A-64
410 IF A=160THEN LET C=32
420 IF A=>161AND A<=191THEN LET C=A-128
430 IF A=>192AND A<=223THEN LET C=A-96
440 IF A=>225AND A<=255 THEN LET C=A-64
450 LET C$=C$+CHR$(C)
460 LET C$=C$+CHR$(146):LET F=F+1
470 GOTO 270
480 REM **********************
```



Dealer Directory

McGILLS AUTHORISED NEWSAGENCY
187 Elizabeth St.,
Melbourne
(602-5566)
COMPUTER BOOKS
DISCOUNT 10%

M.V.B. COMPUTER SUPPLIES
586 Dorset Rd., Croydon
(725-6255)
GOOD PRICES &
FRIENDLY ADVICE

MAXWELL OFFICE EQUIPMENT
162-164 Nicholson St.,
Abbotsford (419-6811)
10% DISCOUNT ON
SOFTWARE

TECHNICAL BOOK & MAGAZINE CO.
295 Swanston St., Melbourne
(663-3951)
Computer Books
(Not Magazines)
& Selected Software
DISCOUNT 10%

MEGATRON COMPUTER INDUSTRIES
62 Charter St, Ringwood
(870-4844)
All Commodore Computers Serviced
10% DISCOUNT ON
LABOUR CHARGE

NOVO COMPUTER HARDWARE
049-201102
P.O. Box 237
Mayfield, NSW, 2304
MEMBER DISCOUNT

COLLINS BOOKSELLERS
401 Swanston St., Melbourne
(654-3144)
COMPUTER BOOKS ONLY
- DISCOUNT 10%

PRESTON ELECTRONIC COMPONENTS
217 High St., Preston
(484-0191)
10% DISCOUNT

TO OBTAIN THE STATED DISCOUNT YOU SHOULD PRODUCE YOUR CURRENT CLUB MEMBERSHIP CARD

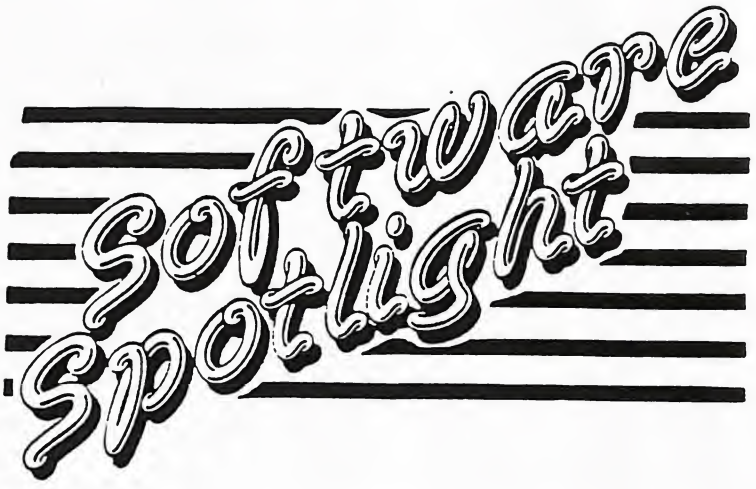
A Chat with the President

Welcome to the club for October. Our September meeting was very well attended, and there were lots of new faces to be seen. I hope that we will see most of them again soon. The committee is always aware that visitors arrive on our doorstep for a great variety of reasons, and unless they leave their first encounter with us feeling comfortable about it, and having achieved something for the evening, we will probably not see them again. This is most likely to occur if nobody talks to them, and they perhaps sit all night listening to a boring presentation about a subject they have absolutely no interest in. You never know when that new person sitting next to you has just solved the problem that has been worrying you for months, or is trying to find out the answer to a question that just happens to be your special subject! The more Commodore users that we can gather together and support each other, the better off we all will be. So next meeting, wear a name tag, and try to find someone new to the club who you want to see again. Communication with other Commodore users is the name of our game.

At this month's meeting we will be returning to a subject that used to be dear to the heart of all computer owners - programming. Now wait! Don't try to run away - we've closed all the doors! It's not something that should be frightening or even hard. There are degrees of programming, from extremely difficult to dead easy, and we will be starting from the easy end. It is not something that owners of modern computers can do very easily, because their machines do not come with a built in programming language. (Although Grant Davies showed us the useful things that can be done on an Amiga last month). But as quite a few of our members own C64s and C128s, the road should be open to further enjoyment of their computers, because these computers have resident a really good version of BASIC (128) or a serviceable one (64). I am not suggesting that we should all sit down and attempt to write the definitive masterpiece for the C64, although some people would get great enjoyment from this.

What I am suggesting is that owners of a C64/128 could use a knowledge of programming to enhance their enjoyment of their computer in different ways. Have you ever acquired a program which promises to do just what you want in printing out text files, or disk directories, or graphics? You load it up, go through the request process, and select "print". Your printer either spews out garbage or just sits there looking cute on the desk. OK, the program is written in BASIC, so let's look at it. You know it works on your friend's printer. All you need is knowledge to change just one line, or even one statement. Don't you wish somebody had given you a clue about what "PRINT#4,CHR$(27)" meant? The same thing applies when an adventure game won't let you pick up the magic wand, even if you have spent all week gathering the keys and killing the evil guards. Frustration!! If you could follow the programming, you could fix the program (and the world) with the greatest of ease. Ask Dorothy!

So we will try to look at programming, starting more or less from the beginning. Those of you who would like to know more about your computer should find it interesting, and those of you who have done programming before might join in the group and contribute your expertise to the newer owners. Both categories should make it known what path through the programming maze we should take, what speed we should go, and what they think should happen. Elaine Foster will be our guide, and I thank her for offering to share her programming knowledge with us. We may eventually get on to machine language coding, or using a monitor or assembler. Once again, not for the purpose of creating a masterpiece, but just to learn the principles involved, and maybe to improve the efficiency of a particular module of a BASIC routine. We will let our course be directed by the needs of members, and I hope that we have not



C64 October Club Disk
A review by Dorothy Millard

HIRES DEMO KIT - Title screen demo maker, includes music - "The Hunter" was my favourite.

SPECIAL DELIVERY - In Special Delivery you are a dedicated postman, charged with carrying a special delivery letter to the crazy hermit who lives in the woods. To complete your assignment you must evade a threatening snake, a booby trap, a persistent dog and a mettlesome rabbit. You must also take care to prevent the rain from soaking the letter. The final obstacle is the crazy old man himself, who might shoot you if he's in a bad mood.

This one brings back memories of typing in listings from magazines and then debugging them, many moons ago! A fairly simple graphics adventure consisting of around 20 locations written in BASIC. The graphics are pretty terrible but the game is fun.

Hints - Climb the gate and feed the rabbit. At the hermit's cabin while carrying the letter knock on the door.

TIC TAC ARITHMETIC - This game was originally written for the PET, which means it has been around for a looooong time. The object is to get three boxes in a row horizontally, vertically or diagonally. You win a box by solving the maths problem in a box. The problems are easy making this game suitable for young children.

DISK RESTORE - This program allows you to restore your formatted disk. "Not possible", I hear you say. Actually it does work but only on disks which have been reformatted and where NO ID is stated when you format a previously formatted disk, which means that only the Block Allocation Map is reconstructed. It is all explained in the instructions which are most informative and comprehensive. I recommend this program.

E-Z SEQ READ - This is a simple sequential file reader. Nothing special.

FILETYPE CONVERT - A utility which makes the conversion to other file types easy. The types catered for include SEQ, USR, REL and DELeted.

MAILCALL V2 - Mailcall is a database to keep track of names, addresses, telephone numbers and birthdays. It is easy to use.

BOB JOGS - A fun animation. "A late jog" - you'll realise why when you see it!

CONVERSIONS CHART MAKER - A very useful program which allows you to print out charts of conversions, ie. Centigrade to Fahrenheit and visa versa, Litres to Gallons, Kilometres to Miles, Centimetres to Inches, Kilograms to Pounds and Grams to Ounces.

STIX - A classic game which looks simple but is very addictive. Uses a joystick. Sometimes the best games are the simple ones!

A Chat with the President (Continued.....)

become just a club of "application users" who only know how to turn the machine on, and blindly accept what a programmer in a far-off place has given us.

I also hope that our members with Amigas or other machines will not feel totally left out of this process. The principles of programming are fairly universal, and what we demonstrate on a C64 applies also to BASIC on the Amiga or any other computer (with modifications of course).

Looking forward to seeing you.

From the Club Shop (alias Pedlar)

BLANK DISKS

5 1/4" - $6 per box of 10
3 1/2" - $9 per box of 10

Please help by bringing the right money if possible.

MEMBERS MICRO MART

FOR SALE

Amiga 2000 - 1.5 MG RAM
with 52 MG Hard Drive
Commodore 1084S Monitor
Large amount of original software ideally suited to someone interested in strategic and roleplaying games
Contact Ken Holt
Flat 13/3 Cape Street, Heidelberg
Telephone (03) 459-4578
$1700 the lot negotiable

FOR SALE

Citizen Printer 5800 9 PIN $100
(2 Manuals) 5800 + 120D

Amiga Video Titler HT 100 (3 disks + manual) - Requires 1 MG RAM $199

Contact Scott Sears,
35 Francis Street, Blackburn
Telephone (03) 877-4829

Members' Ads. relating to computing are free in the newsletter.

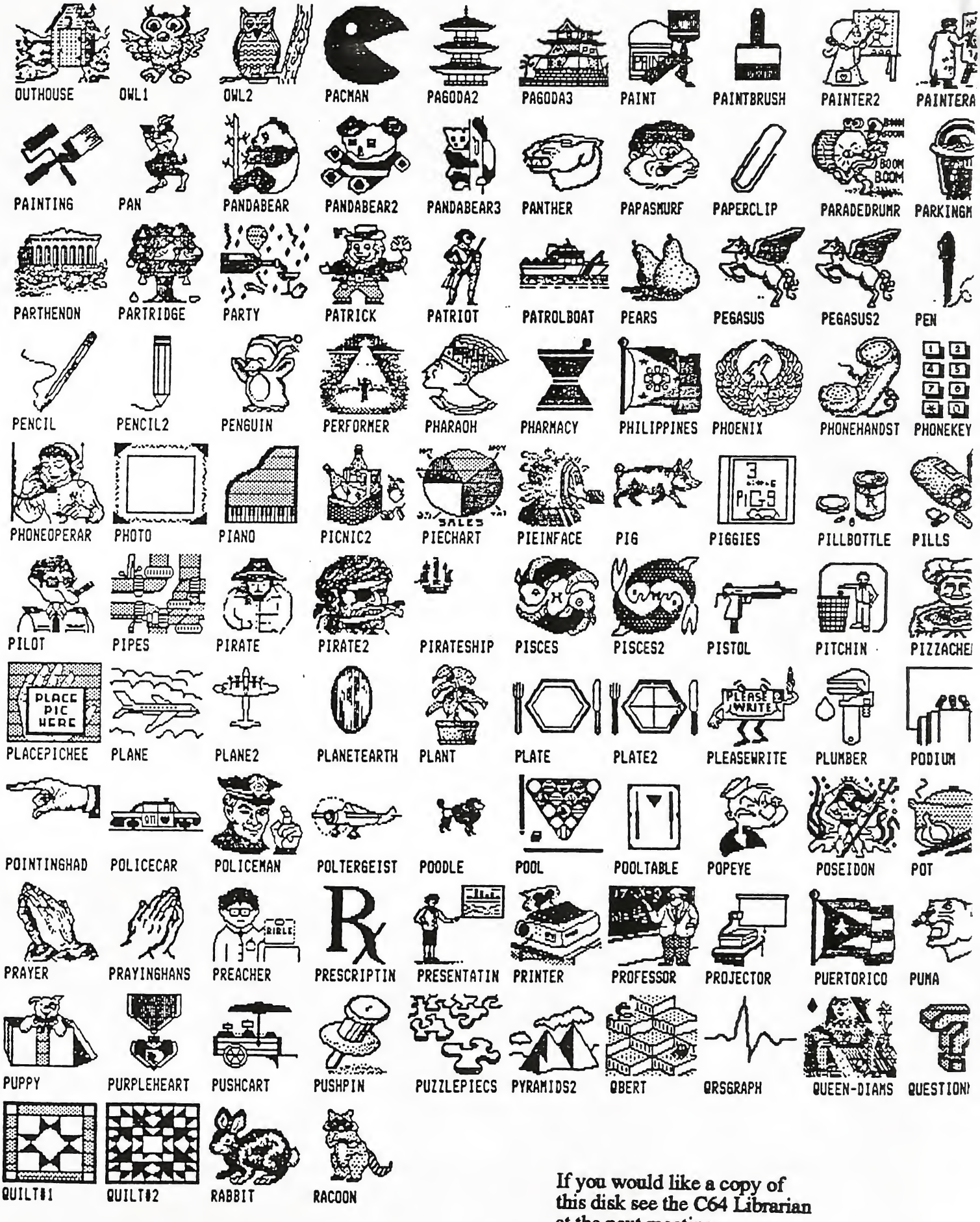
C64 Public Domain Software (Continued.....)

COLOUR SPELLING TEST - When a colour appears on the border and the computer speaks that colour you must spell the colour correctly, then the borders will flash. This is a very simple short game suitable for teaching young children how to spell and recognise colours.

Conclusion - With the heaps of different programs on this disk there has to be something for everyone. Don't miss it - it has to be good value for just $2.

Club disks are available from the C64 Librarian at the club meeting for only $2. They arealso available through the mail for the same price plus postage.

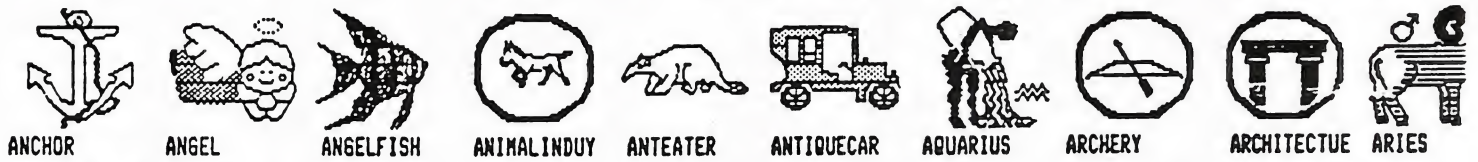
PRINTSHOP GRAPHICS ON MCCC AUGUST 94,1B



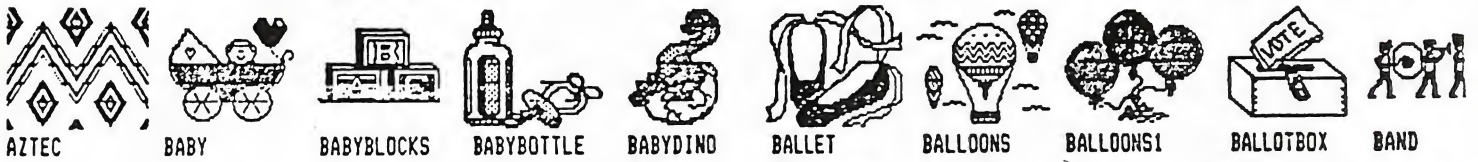
If you would like a copy of this disk see the C64 Librarian at the next meeting.

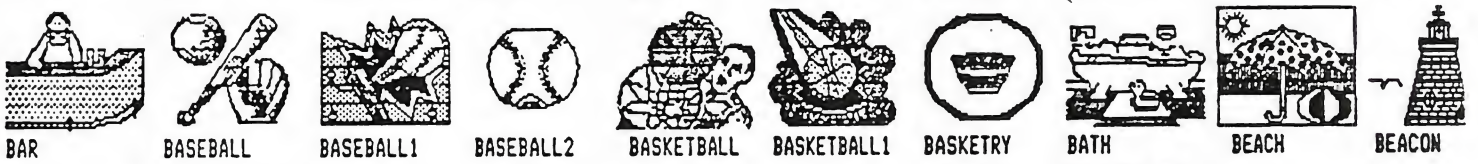
PRINTSHOP GRAPHICS ON MCCC AUGUST 94, 1A

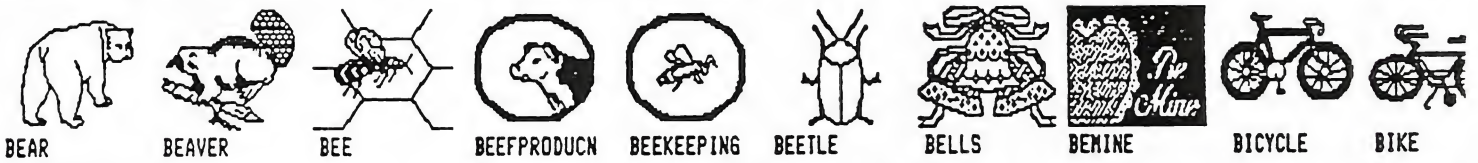

2HANDS1HEAT ABE ABLOCK AGRICULTURB AIRPLANE ALARMCLOCK ALBATROSE ALIGATOR ALOHA AMERICANE


ANCHOR ANGEL ANGELFISH ANIMALINDUY ANTEATER ANTIQUECAR AQUARIUS ARCHERY ARCHITECTUE ARIES

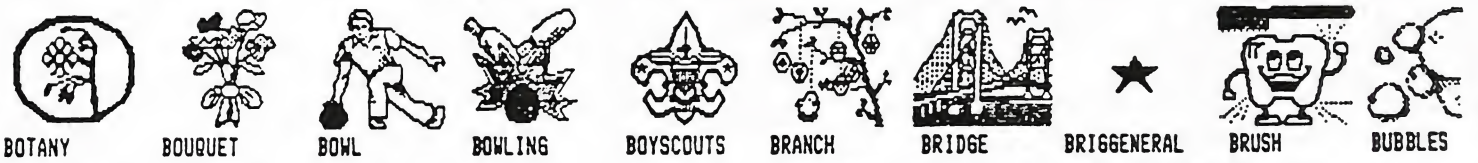

ART ART1 ARTIST ASTRONOMY ATHLETICS ATLAS ATOM ATOMICENERY AUTOMOTIVEE AVIATION


AZTEC BABY BABYBLOCKS BABYBOTTLE BABYDINO BALLET BALLOONS BALLOONS1 BALLOTBOX BAND

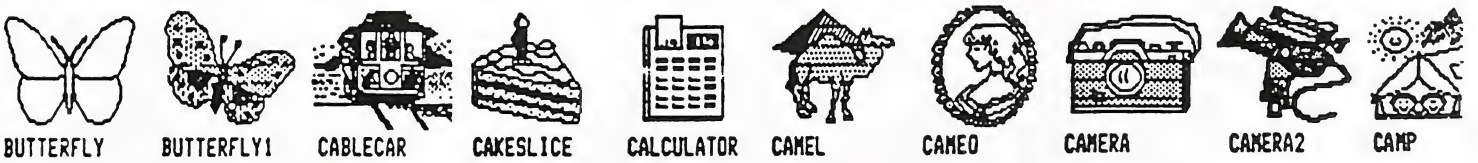

BAR BASEBALL BASEBALL1 BASEBALL2 BASKETBALL BASKETBALL1 BASKETRY BATH BEACH BEACON


BEAR BEAVER BEE BEEFPRODUCN BEEKEEPING BEETLE BELLS BEMINE BICYCLE BIKE

BINGO BIRDHOUSE BIRDSTUDY BIRTHDAY BIRTHDAYCAKE BLOSSOM BOOKBINDING BOOKS BOOKS1 BOOTS


BOTANY BOUQUET BOWL BOWLING BOYSCOUTS BRANCH BRIDGE BRIGGENERAL BRUSH BUBBLES


BUBBLY BUGLING BULBS BULL BULLDOG BULLFROG BUNNY BURGER BUS BUTTERFL


BUTTERFLY BUTTERFLY1 CABLECAR CAKESLICE CALCULATOR CAMEL CAMEO CAMERA CAMERA2 CAMP


CAMPING CANCER CANDLE CANDLESTICK

If you would like a copy of this disk see the C64 Librarian at the next meeting.

Corrupt Practices

● *Burghard-Henry Lehmann*

Amiga disks are notoriously fragile. Burghard-Henry Lehmann explains why disk errors, however, need not be terminal

■ **Amiga disks can easily be corrupted.**

All you have to do is take the disk out of the drive while the green light is still on and before you know it, you've got a hard error, as it is called, on the disk. This results in the disk not being accepted any more by AmigaDOS and if you put it into the drive you'll get a system error report.

This, however, does not mean that the whole disk is unusable. Most of the time it means that only one or two blocks of the disk have become corrupted.

To mend a disk like this, CLI on Workbench 1.2 has a facility called 'Diskdoctor'. This tells you on which cylinder the hard error is and patches it up. But as a result you will have lost the file where the error was. All you can salvage with Diskdoctor are the files which have not been corrupted. A better Way to repair a corrupted disk is "by hand", with a disk editor.

A disk editor is a program that gives you direct access to the data contained on a disk and lets you change it. With most disk editors you choose a block you want to look at and then the disk editor displays the data contained in this block on the screen. You may then change this data in any way you like and get the disk editor to write it back into the appropriate block on the disk.

A disk editor is also useful if you have deleted a file by mistake. If you notice this in time and nothing new has been written onto the disk you may be able to restore the file.

But for this you have of course to know what you are doing, otherwise you might mess up a disk completely ! In this article I'd like to give you the background information which you'll need to access and change a disk successfully. If you haven't got a disk editor, don't despair ! Our public domain librarian can provide you with one.

The Structure of an Amiga disk

Before you can use a 3.5-inch disk on your Amiga it has to be formatted or initialised by Amiga DOS, the disk operating system of the Amiga. AmigaDOS formats a disk into 8O cylinders, 160 tracks and 1760 blocks.

The 80 cylinders are numbered from 0 to 79. Each cylinder consists of 2 tracks - the so called "top track" on one side of the disk and the "bottom track" on the other side, since all disks used on the Amiga are double sided. This means that there is a total of 160 tracks on an Amiga disk.

Each track in turn is formatted into 11 sectors, numbered from 0 to 10. This gives a grand total of 1760 blocks (80 * 2 * 11) on each disk, which are numbered from 0 to 1 759.

When accessing and changing a disk directly with a disk editor you look at a certain block. So it is the blocks that interest us most.

Blocks

Each block holds 512 bytes of data. From this you can calculate that an Amiga disk can hold a total of 901120 bytes of data, even though only about 520000 bytes of this contains pure data, the rest is used to manage the disk and the filing system. Incidentally, if you want to make maximum use of the capacity of an Amiga disk, use few or no directories. This saves on filing information and frees therefore more disk space for data proper !

These 512 bytes are portrayed by most disk editors on the screen in chunks of 4 bytes or "long words", as they are also called. A long word is 32 binary bits long (because of this it is also called a "32-bit number"). Therefore, each block contains 128 long words (512/4).

To explain this a bit further, let's assume a long word is pointing at block 350 of the disk. Most disk editors portray the data contained in a block in hexadecimal because this is more convenient than decimal. Decimal 350 is hex 015E, and since it is shown in form of a long word, block 350 would

Figure 1 : Layout of the Root Block

0	00000002	Type .
1	00000000	Header key (always zero)
2	00000000	Highest sequential number (always zero)
3	00000048	Hash table size (always 72)
4	00000000	Unused
5	nnnnnnnn	Checksum
6	nnnnnnnn	Hash table
Size-51	nnnnnnnn	
Size-50	FFFFFFFF	Bitmappflag (-1 if Bipmap is valid)
Size-49	nnnnnnnn	Pointer to blocks containing the Bitmap
Size-24	nnnnnnnn	
Size-23	DAYS	Date and time when disk has been altered last
Size-22	MINS	
Size-21	TICKS	
Size-20	DISK	Name of disk in ASCII
Size-8	NAME	
Size-7	CRE-DAYS	Date and time when disk has been created.
Size-6	CRE-MINS	
Size-5	CRE-TICKS	
Size.4	00000000	unused
Size-3	00000000	unused
Size-2	00000000	unused
Size-1	00000000	Secondary block identification

be portrayed as 0000015E.

Conversely, a long word can hold 4 ASCII characters. For example, long word 6469736B reads as "disk" (hex 64 is ASCII "d", hex 69 is ASCII "i" and so on).

To find a certain position on a block with a minimum of counting, one roughly halves the block and counts both from the beginning and the end of the block, whichever is nearer. For this purpose in layout diagrams like figure 1 to 4 the first long word is given as 0 the second as 1, the third as 2 and so on, while the last long word is given as Size-1, the one before that as Size-2, the one before that as Size-3 and so on.

Since the Amiga disk filing system consists of directories, sub-directories and data files, there are five different types of "filing system blocks", as they are called:

* The Root Block
* The User Directory Block
* The File Header Block
* The File List Block
* The Data Block

Furthermore, there are two types of blocks which are of equal importance to the management of a disk, but which are only mentioned in very advanced literature:

* The Boot Block
* The Bitmap

The two Boot Blocks, which are always block 0 and 1, are of little interest to you, since they are only used to boot a disk. But the Bitmap, which tells the DOS which sectors on the disk are used and which ones are free, is quite important. That's why we will have a close look at it in a minute

The Filing System

The layout of the five types of filing system blocks is quite consistent and can therefore easily be grasped:

The first six long words (0 to 5) contain the header information and the checksum. This is also called the primary block identification and it tells you what kind of block you are looking at.

Next there are 72 long words which contain the hash table. The hash table holds the block numbers of the User Directory Blocks, File Header Blocks and Data Blocks the block is pointing at.

Next there are 27 long words (Size-50 to Size-24) which either contain the Bitmap pages, if it is the Root Block, or the protection flags and the text of the directory or file comment if it is a User Directory Block or a File Header Block.

The next 19 long words (Size-23 to Size-5) which give the creation date of the disk or directory or file and its name.

Finally there are four long words (Size-4 to Size-1) which give the so-called secondary block identification.

The Root Block

The Root Block is, apart from the two Boot Blocks, the only fixed block of a disk. It is always block 880, that is cylinder 40, track 0, sector 0. Thus the Root Block lies exactly in the center of the disk. All the other blocks are installed in a fluid manner as the disk is filled up and its contents are changed.

To find your way around a disk with a disk editor, you first call up the Root Block (most disk editors do this automatically when you pop a new disk into the drive). The Root Block tells you where the first generation User Directory and File Header Blocks are and where the Bitmap is. In turn, the first generation User Directory Blocks tell you where the second generation User Directory and File Header Blocks are, and finally, the File Header Blocks point to the File List and the Data Blocks and the Data Blocks which contain the data itself which is stored on the disk.

Let's now look a bit closer at the layout of the Root Block (see figure 1): The first long word gives you the block type number which is 2. (This is the same for the User Directory Blocks, the File Header Blocks and the File List Blocks. Only the Data Blocks have a different block type number, which is 8.) The next two long words (1 and 2) of the Root Block are always zero because they are not used. The fourth long word (3) gives the size of the hash table which is always 72 (hex 48). The sixth long word (5) contains the checksum. (Note that 5 always contains the checksum on any of the filing system blocks !) The checksum is the sum of all the long words in the block, expressed in a negative number. For those of you who, like me, are none the wiser after a statement like this, let me give you an example:

Imagine that all the long words in a block are 00000000, that is the whole block contains zero. In this case the checksum too is 00000000. Now one long word is 00000001. In this case the checksum is FFFFFFFF (decimal -1). If another long word would now contain 00000002 then the checksum would be FFFFFFFD (decimal -3), that is 00000000 - (00000001 + 00000002). The important thing about the checksum of any block is that it has to be correct. If it isn't, as happens frequently when a block has become corrupted, then the disk is considered to be faulty by the DOS and not accepted. Most disk editors allow you to mend this easily by calculating the correct checksum for you.

The Hash table

The next 72 long words contain the hash table, that is the block numbers of the User Directory Blocks and File Header Blocks the Root Block is pointing at.

It is called the hash table because the entries contained in it are sorted in a special way which is called "hashing" or "applying a hash function".

Whenever you have a large number of data which you want the computer to search you can employ different methods of sorting that data. The best known way with alphanumeric or string data is of course sorting it alphabetically, like in a dictionary. In this case the "hash function" (or the sorting algorithm) which you apply works like this: All words starting with the letter 'a' are given the hash code 1,

Figure 1 : Layout of User Directory Block

0	00000002	Type .
1	OWN KEY	Header key (points to self)
2	00000000	Highest sequential number (always zero)
3	00000048	Hashtable size (always 72)
4	00000000	Unused
5	nnnnnnnn	Checksum
6	nnnnnnnn	Hashtable
Size-51	nnnnnnnn	
Size-50	00000000	unused
Size-48	PROTECT	Protection bits
Size-47	00000000	unused
Size-46	nnnnnnnn	Comment string stored in ASCII
Size-24	nnnnnnnn	
Size-23	DAYS	Creation date and time
Size-22	MINS	
Size-21	TICKS	
Size-20	DIRECTORY	Name of directory in ASCII
Size-5	NAME	
Size.4	nnnnnnnn	Hashchain pointer (zero if hashchain finished)
Size-3	nnnnnnnn	Back pointer to parent directory
Size-2	00000000	unused
Size-1	00000002	Secondary block identification

all words starting with 'b' are given the hash code 2, and so on. Hash code 1 is of course the first position in the hash table, while hash code 2 is the second position.

Now all words which start with the same letter and therefore yield the same hash code are put into what is called a "hash chain". That is, each word in the chain has a pointer which points to the next word in the chain. If there are no more words in this chain, the pointer contains zero, meaning that this particular hash chain is finished"

The idea of hashing is of course to come up with a hash function which makes the search through the data as efficient and speedy as possible. This means, you want to prevent "bunching", where most data is confined in one or two hash chains which are very long

Unfortunately I haven't got the information as to which hash function AmigaDOS uses to chain directories and files. It certainly doesn't do it alphabetically !

Nevertheless, the directories and files contained in a disk are chained in the following way: Let's say position 10 of the hash table of the Root Block points at block 883. This might be a User Directory or a File Header and it is the first link in the hash chain with hash code 10. To find the next link in that hash chain you look at Size-4 of that User Directory or File Header Block. If this contains zero then the chain ends with this block. Otherwise the hash chain continues with the block whose number is given in Size-4.

Therefore, when Amiga-DOS searches for a particular directory or file it first uses the hash function on the name to extract the hash code. Then it searches through all the names in that hash chain. If a name matches with the searching name of the directory or file it is it has found the User Directory Block or File Header Block of that directory or file. If it comes to the end of the hash chain, that means that the search has been unsuccessful.

Next the Root Block tells you where the Bitmap is and if it is valid.

If the Bitmap is valid, Size-50 (remember, this is counted backwards from the end of the block !) contains FFFFFFFF (that is -1). Size-49 to Size-24 contains the Bitmap pages, even though I haven't seen a disk yet which contains than more than one Bitmap.

Next Size-23 to Size-21 contains the time, that is, the days, the minutes and the ticks, when the disk has been altered last. After this (Size 20 to Size 8) comes the name of the disk given in ASCII. Then the original creation date of the disk is given (Size-7 to Size-5)

Size-4 to Size-2 is not used on the Root Block and therefore contains zero.

Finally, Size-1 or the last long word of the Root Block contains the secondary block type number which is 1.

The User Directory

If there are any directories or subdirectories on the disk then there is a

Figure 3 : Layout of File Header Block

0	00000002	Type .
1	OWN KEY	Header key (points to self)
2	nnnnnnnn	Total number of data-blocks in file
3	nnnnnnnn	Number of data block slots used
4	nnnnnnnn	First data block
5	nnnnnnnn	Checksum
6	DATA B-n	List of data blocks
	DATA B-3	
	DATA B-2	
Size-51	DATA B-1	
Size-50	00000000	unused
Size-48	PROTECT	Protection bits
Size-47	nnnnnnnn	Total size of file in bytes
Size-46	nnnnnnnn	Comment string stored in ASCII
Size-24	nnnnnnnn	
Size-23	DAYS	Creation date and time
Size-22	MINS	
Size-21	TICKS	
Size-20	FILE	Name of file in ASCII
Size-5	NAME	
Size.4	nnnnnnnn	Hash chain pointer (zero if hash chain finished)
Size-3	nnnnnnnn	Back pointer to parent directory
Size-2	nnnnnnnn	Pointer to first extension block (zero if none)
Size-1	FFFFFFFD	Secondary block identification

Figure 4 : Layout of File List Block

0	00000002	Type .
1	OWN KEY	Header key (points to self)
2	nnnnnnnn	Number of data blocks in block list
3	nnnnnnnn	Same as above
4	nnnnnnnn	First data block
5	nnnnnnnn	Checksum
6	BLOCK n+n	Extended list of data block keys
	DATA n+3	
	DATA n+2	
Size-51	DATA n+1	
Size-50	00000000	unused
Size-5	00000000	unused
Size.4	00000000	unused
Size-3	nnnnnnnn	Back pointer to File Header Block
Size-2	nnnnnnnn	Pointer to next extension block (zero if none)
Size-1	FFFFFFFD	Secondary block identification

Figure 5 : Layout of Data Block

0	00000008	Type .
1	OWN KEY	Header key (points to self)
2	nnnnnnnn	Sequential number
3	nnnnnnnn	Size of data
4	nnnnnnnn	Next data block
5	nnnnnnnn	Checksum
6	nnnnnnnn	DATA

User Dictionary Block for every one of these.

As you can see from figure 2 the layout of the User Dictionary Block is rather similar to the layout of the Root Blocks.

The main differences are in the bottom of the block: Size-4 contains now the pointer to the next link in the hash chain (or zero, if the hash chain is finished).

Size-3 points back to the parent directory or the Root Block.

The File Header Block

The File Header Block (see figure 3) heads a file which is stored on the disk. If it is a large file with more than 72 Data Blocks then there is a pointer to a File List Block which serves as an extension of the File Header Block. Otherwise the Data Blocks are listed where the hash table is on the Root Block and the User Directory Block.

If there is an extension, Size-2 points to the first File List Block.

The secondary block identification number of a File Header Block (as well as a File List Block) is FFFFFFFD (decimal -2).

The File List Block

As I 've said above, a File List Block (see figure 4) is employed if a file is larger than 72 Data Blocks. If one File List Block is not sufficient to point to the rest of the Data Block there is a further File List Block whose block number is given in Size-2, the extension pointer.

The Data Block

Finally we come to the Data Block (see figure 5) which contains the data stored in a file itself.

The Data Block uses only the first six long words for filing information. The rest of the block is free for the data proper. Therefore, each Data Block can contain up to 488 bytes of data.

To chain the Data Blocks of a file in the right order, all Data Blocks are given a sequential number starting from 1. This number is contained in position 2.

The Bitmap

In order to store new data on a disk AmigaDOS has to know which blocks are used and which ones are free. This information is stored in a special block, called the Bitmap.

The Bitmap really is very simple: Each block on the disk is represented by a binary digit. If the bit is on (=1) then the block it represents is free. If the bit is off (=0) then the block is used.

The first long word of the Bitmap contains the checksum. Then the Bitmap itself starts. But it starts with block 2, since block 0 and 1 are always used as the two Boot Blocks (as a matter of fact, blocks 0 and 1 are appended at the end of the Bitmap). Thus the second long word contains the allocation information for block 2 to 33, while the third long word contains the information for block 34 to 65 and so on.

To find a block on the Bitmap you first have to find the long word or position where the block is included. To do this subtract 2 from the block, divide the number of the block by 32 and round the result up to its nearest integer; if the result is an integer, add 1. For example, 2-2/32=0, 0+1 gives position 1; 34-2/32=1, 1+1 gives position 2; 880-2/32=27.43, rounded up to nearest integer gives position 28 and so on.

As a matter of fact, on a disk which has been newly initialised on Workbench, naturally, most of the blocks on such a disk are free, which means that most long words contain FFFFFFFF. Only in the center part of the disk five blocks are used (the Root Block, the Trashcan Directory) and the Trashcan Information.

The above article on Corrupt Practices, written by Burghard-Henry Lehmann, has been reprinted from ESEUC NEWS, with thanks.

Melbourne Commodore Computer Club Inc.

If undeliverable return to:

P.O. Box 177,
BOX HILL, VIC. 3128.

C LUB MEETING DATES FOR 1994/95

12th January, 1994.	9th February, 1994.	9th March, 1994.
13th April, 1994.	11th May, 1994.	8th June, 1994.
13th July, 1994.	10th August, 1994.	14th September, 1994.
12th October, 1994.	9th November, 1994.	14th December, 1994.
11th January, 1995	8th February, 1995.	8th March, 1995.

Please Note:
All club meetings are on the second Wednesday of each month in the Courtyard Room, Nunawading Civic Centre, Whitehorse Road, Nunawading. The meeting room is available between 7-11 p.m.